



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/797,238	03/10/2004	Rajeev B. Rajan	MSFT-2924/306986.01	2995
41505 7590 12/04/2008 WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION) CIRA CENTRE, 12TH FLOOR 2929 ARCH STREET PHILADELPHIA, PA 19104-2891				
EXAMINER				
TIMBLIN, ROBERT M				
ART UNIT		PAPER NUMBER		
2167				
MAIL DATE		DELIVERY MODE		
12/04/2008		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/797,238

Applicant(s)

RAJAN ET AL

Examiner

ROBERT TIMBLIN

Art Unit

2167

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 9/15/2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-12, 14-17, 19-22 and 24-35 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-12, 14-17, 19-22 and 24-35 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/888)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

This Office Action corresponds to application 10/797,238 filed 3/10/2004.

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/15/2008 has been entered.

Response to Amendment

In the present reply (amendments submitted 9/15/2008) Applicant has amended claims 1-6, 14, 19, and 24. Although Applicant remarks claims 1-28 to be pending (see "Remarks", page 9 in the reply), claims 1-12, 14-17, 19-22, and 24-35 are currently noted as pending and are under examination as no further claim cancellations have been made.

Claim Objections

Examiner thanks Applicant for the correcting amendments to overcome the objections. Accordingly, the previous claim objections have been withdrawn.

However, upon further examination, the following present claim objection is noted:

With respect to claim 1, line 11 of this claim is objected to because "at last one query..."should be corrected to read "at least one query..."

Response to Arguments

Applicant's arguments filed in the reply dated 9/15/2008 (i.e. 'reply' hereafter) have been fully considered but they are not persuasive.

Applicant argues on page 13 of the reply that Sedlar fails to teach wherein each item in a statement is stored separately from a database table associated with an item.

As noted in the Office Action below, the Examiner respectfully submits that this element is taught. For example, Sedlar teaches Sedlar teaches storing a hierarchical index separately from a files table (at least in figures 5 and 7 and col. 10, line 8-23). For this example, Sedlar describes a hierarchical index for locating entries (e.g. see also figure 8) while files table 710 store the files as a BLOB. Accordingly, items to be retrieved are referenced in the hierarchical index and therefore are stored separately.

Applicant also argues (page 13 of the reply) that Sedlar does not teach receiving each of at least one query language statement in that Sedlar fails to teach handling both file system statements and query language statements.

The Examiner respectfully disagrees because Sedlar teaches querying a database using a database language (SQL) in col. 4 line 66-67. Sedlar further teaches the handling file system statements (e.g. file system calls) in col. 12 line 45-46. Furthermore, Sedlar teaches a mapping of file system commands to relational commands in col. 13 line 1-10 to suggest handling both types of statements (query language and file system). Accordingly, by disclosing a system that handles both file system calls and relational operators, Sedlar teaches this argued limitation.

Applicant further argues (page 13 of the reply) that Sedlar does not teach creating a transaction, wherein the transaction is managed separately from the database server. The Examiner disagrees as this element is taught (col. 12 line 48-61, figure 3) as an OS File API controlling the transaction. Specifically, Sedlar teaches that this API includes routines to perform operations such as open, read, write, lock, and close (col. 12 line 62-67). The Examiner submits that because the OS file API controls the functions typically considered as transaction routines, that Sedlar teaches managing a transaction separately from the database server. In other words, the OS file API manages the transaction rather than the database server 204 (see figure 3).

Finally, Applicant argues (page 13 of the reply) that Sedlar fails to teach for each query language statement, starting a transaction on the database server updating fields associated with the query language statement and sending an updategram to the database server.

The Examiner submits that arguments to this limitation are moot in view of the new interpretation of the Sedlar reference. Specifically, the Examiner submits that this limitation is taught by Sedlar substantially in the cited portions of the Office Action below. For example, Sedlar teaches a series of operations ((1)-(6) in col. 14 line 10-16) that causes an update to be performed and thus teaches the claimed "updategram".

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1, 14, 19, and depending claims therefrom are rejected under 35 U.S.C. 101 because they are directed towards non statutory subject matter.

Specifically, these claims recite a method which is not tied to another statutory category or transforms a particular article into a different state or thing. Accordingly, under recent court decisions (e.g. refer In re *Bilski*) a process must pass the machine-or-transformation test. In this test, the supposed process is patent eligible under section 101 if: (1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing. Further, the involvement of the machine or transformation in the claimed process must not merely be insignificant extra-solution activity; and that the transformation must be central to the purpose of the claimed process.

Currently, the claims s should recite structural elements of machine hardware to tie the methods to a particular machine or apparatus. Further, the recited hardware should be actively recited in a specified manner as to impose meaningful limits on the claim's scope to impart patent-eligibility.

Claim 24 and depending claims are now accepted under 35 USC 101 for claiming a hardware system (i.e. a machine) including a processor. In accordance with figure 7 and Applicant's paragraph 0059, the processor is best construed as a hardware processor in a computer system (i.e. it is connected to various hardware components in a computer). The rejection has been withdrawn.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1, 3-12, 14, 16-17, 20-22, 24, and 26-35 are rejected under 35 U.S.C. 102(e) as being anticipated by Sedlar U.S. Patent 6,922,708. In the following citations, drawings, and drawing references, Sedlar teaches:

With respect to claim 1, A method for executing a transaction comprising at least one query language statement and at least one file system statement, each statement relating to a user defined type ("UDT") associated with a database server, the method comprising:

receiving each of the at least one file system statements (col. 17 line 29; e.g. transaction statement), wherein each statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) comprises a call to open an item (col. 12 line 65 – opening a file) and at least one of a call to read from the item (col. 12 line 65 – reading a file) and to write to the item (col. 12 line 65 – writing to a file), and a call to close the item (col. 12 line 66 – closing a file), wherein each item reference in a statement is stored separately (drawing reference 710) from a database table (drawing reference 510) associated with the item (col. 10 line 8-23, figs 5,7; e.g. Sedlar teaches storing a hierarchical index separately from a files table);

receiving each of the at last one query language statement (col. 6 line 25-28; e.g. receiving a query), wherein each query language statement (col. 6 line 25-28; e.g. receiving a query) is associated with an item (col. 6 line 29-33; e.g. a file);

for each file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call):

if the file system statement includes open, read and close operations (col. 13 line 3-10);

creating a transaction (col. 13 line 3-4; e.g. beginning a transaction) as part of an open operation (col. 13 line 3), wherein the transaction is managed separately (col. 12 line 48-61, figure 3; e.g. OS File API) from the database server (drawing reference 204);

obtaining a read lock on a data table row (col. 14 line 1-4) associated with the associated item (col. 13 line 5-10; e.g. a file) for the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call);

performing a read operation in the context of the transaction (col. 13 line 6; e.g. read from a file);

committing the transaction as part of a close operation (col. 13 line 17-18);

if the file system statement includes open, write and close operations (col. 13 line 5-10);

creating a transaction (col. 13 line 3-4; e.g. beginning a transaction) as part of an open operation (col. 13 line 3), wherein the transaction is managed separately (col. 12 line 48-61, figure 3; e.g. OS File API) from the database server (204);

obtaining a write lock (col. 13 line 36-41) on a data table row associated with the associated item (file) for the file system statement (col. 17 line 29);

performing a write operation in the context of the transaction (col. 13 line 5);

committing the transaction as part of a close operation (col. 13 line 17-18);

for each query language statement (col. 6 line 25-28; e.g. receiving a query), starting a transaction (col. 14 line 12; operation (1)) on the database server (204) updating fields

(col. 14 line 14-15; operation (5)) associated with the item (file) in the query language statement (col. 6 line 25-28; e.g. receiving a query) and sending an updategram (col. 14 line 5-22) to the database server (204);

With respect to claim 3, the method of claim 1, wherein each query language statement is a T-SQL statement (col. 13 line 1-10).

With respect to claim 4, the method of claim 1, wherein transactions created for the file system statements are managed by a storage platform (col. 12 line 48-61 and line 62-67, figure 3; e.g. OS File API).

With respect to claim 5, the method of claim 1, further comprising receiving a transaction context for file system operations (col. 10 line 55-56) and performing at least one of a read lock and a write lock consistent with the received transaction context (col. 13 line 7 and col. 14 line 1-5).

With respect to claim 6, the method of claim 1, wherein creating the transaction comprises:

determining whether creating the transaction will result in a conflict (col. 13 line 35-46; e.g. determining a session lock on data);

if creating the transaction will result in a conflict, then resolving the conflict according to a conflict resolution scheme (col. 13 line 36-40 and col. 14 line 1-5; e.g. waiting until a transaction commits to see changes to data or new rows); and

if creating the transaction will not result in a conflict, then starting the transaction (col. 13 line 42-46).

With respect to claim 7, the method of claim 1, wherein acquiring the read lock on the row comprises acquiring a read committed view of the row (col. 14 line 4-5).

With respect to claim 8, the method of claim 1, wherein acquiring the write lock on the row comprises acquiring a write lock that will prevent another transaction from accessing the row while the transaction is being processed (col. 13 line 35-47).

With respect to claim 9 the method of claim 1, wherein acquiring the write lock on the row comprises acquiring a write lock that will prevent a non-transacted file system statement from accessing the row while the transaction is being processed (col. 14 line 5-22).

With respect to claim 10, the method of claim 1, wherein acquiring the write lock on the row comprises acquiring a write lock that will prevent another statement within the transaction from writing to the row (col. 13 line 66-col. 14 line 2).

With respect to claim 11, the method of claim 1, wherein acquiring the write lock on the row comprises acquiring a write lock that will enable another statement within the transaction to read from the row (col. 13 line 56-57).

With respect to claim 12 and similar claims 17, 22 and 35, the method of claim 1, comprising starting the transaction by acquiring one of a read lock and a write lock on a file stream field of the row (col. 10 line 8-24 and col. 11 line 41).

With respect to claim 14, A method for locking and isolation of a file system statement, the method comprising:

receiving the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) comprising a call to open an item (col. 12 line 65 – opening a file), a call to read from the item (col. 12 line 65 – reading a file) or to write to the item (col. 12 line 65 – writing to a file), and a call to close the item (col. 12 line 66 – closing a file), the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) being independent of any database commands employing a query language of a database (col. 5 line 10-13) and wherein each item reference in a statement is stored

separately (drawing reference 710) from a database table (drawing reference 510) associated with the item (col. 10 line 8-23, figs 5,7; e.g. Sedlar teaches storing a hierarchical index separately from a files table);

in response to receiving the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) that is independent of any database application programming interface requests (col. 5 line 10-13), determining a read lock is available (col. 13 line 35-45) for a row of a data table corresponding to the item (col. 13 line 7 and 41-42 – i.e. locking a row associated with a file);

if the read lock is not available for the row of the data table corresponding to the item, then failing the open (col. 13 line 36-40 and col. 14 line 1-5; e.g. waiting until a transaction commits to see changes to data or new rows); and

if the read lock is available for the row of the data table corresponding to the item:

performing a shared open for the item (col. 13 line 54-58);

acquiring a read lock on the row (col. 13 line 42-46).

With respect to claim 16, the method of claim 14, wherein acquiring the read lock on the row comprises acquiring a read committed view of the row (col. 14 line 4-5).

With respect to claim 19, A method for locking and isolation of a file system statement, the method comprising:

receiving the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) comprising a call to open an item (col. 12 line 65 – opening a file), a call to read from the item (col. 12 line 65 – reading a file) or to write to the item (col. 12 line 65 – writing to a file), and a call to close the item (col. 12 line 66 – closing a file), the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) being independent of any database commands employing a query language of a database (col. 5 line 10-13) and wherein each item reference in a statement is stored separately (drawing reference 710) from a database table (drawing reference 510) associated with the item (col. 10 line 8-23, figs 5,7; e.g. Sedlar teaches storing a hierarchical index separately from a files table);

in response to receiving the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) that is independent of any database application programming interface requests (col. 5 line 10-13), determining if a write lock is available (col. 13 line 35-45) for a row of a data table corresponding to the item (col. 13 line 7 and 41-42 – i.e. locking a row associated with a file);

if the write lock is not available for the row of the data table corresponding to the item, then failing the open (col. 13 line 36-40 and col. 14 line 1-5; e.g. waiting until a transaction commits); and

if the write lock is available for the row of the data table corresponding to the item:

performing an exclusive open for the item (col. 14 line 1-3);

acquiring the write lock on the row (col. 13 line 42-46).

With respect to claim 21, the method of claim 19, wherein acquiring the write lock on the row comprises acquiring a write lock that will prevent another statement from accessing the row while the statement is being processed (col. 13 line 40-45).

With respect to claim 24, A system for executing a file system statement in the context of a transaction, the file system statement including a call to open an item, one of a call to read from the item and a call to write to the item, and a call to close the item, the system comprising:

a processor (1804);

a relational data engine (figure 7 and 204) comprising a data table (710) having a row (fig. 7, Row ID) corresponding to the item (File ID);

a storage platform (figures 3 and 4, references 108 and 204) built on the relational data engine (figure 7 and 204), the storage platform (figures 3 and 4, references 108 and 204) comprising means for receiving the file system statement (figure 3), wherein each item reference in a statement is stored separately (drawing reference 710) from a

database table (drawing reference 510) associated with the item (col. 10 line 8-23, figs 5,7; e.g. Sedlar teaches storing a hierarchical index separately from a files table), means for associating the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) with the transaction (col. 3 line 43-46), and means for starting the transaction by acquiring one of a read lock (col. 14 line 1-3) and a write lock (col. 12 line 65-66) on a data table row ((col. 13 line 7 and 41-42 – i.e. locking a row associated with a file) the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) comprising a call to open (col. 12 line 65 – opening a file), a call to read from the item (col. 12 line 65 – reading a file) or to write to the item (col. 12 line 65 – writing to a file), and a call to close the item (col. 12 line 66 – closing a file), the file system statement (col. 3 line 40-45 and col. 12 line 45-50; i.e. an OS file system call) being independent of any database commands employing a query language of a database (col. 5 line 10-13).

With respect to claim 26, the system of claim 24, wherein the storage platform further comprises means for associating a second statement with the transaction (col. 13 line 59-64).

With respect to claim 27, the system of claim 26, wherein the second statement is another file system statement (col. 13 line 65-67).

With respect to claim 28 the system of claim 26, wherein the second statement is a transactional query language statement (col. 14 line 10-20).

With respect to claim 29, the system of claim 24, wherein the means for starting the transaction comprises means for performing the following steps:

determining whether starting the transaction will result in a conflict (col. 13 line 35-46; e.g. determining a session lock on data);

if starting the transaction will result in a conflict, then resolving the conflict according to a conflict resolution scheme (col. 13 line 36-40 and col. 14 line 1-5; e.g. waiting until a transaction commits to see changes to data or new rows); and

if starting the transaction will not result in a conflict, then starting the transaction (col. 13 line 42-46).

With respect to claim 30, the system of claim 24, wherein the read lock provides a read committed view of the row (col. 14 line 4-5).

With respect to claim 31, the system of claim 24, wherein the write lock prevents another transaction from accessing the row while the transaction is being processed (col. 13 line 35-47).

With respect to claim 32, the system of claim 24, wherein the write lock prevents a non-transacted file system statement from accessing the row while the transaction is being processed (col. 14 line 5-22).

With respect to claim 33, the system of claim 24, wherein the write lock prevents another statement within the transaction from writing to the row (col. 13 line 66-col. 14 line 2).

With respect to claim 34, the system of claim 24, wherein the write lock enables another statement within the transaction to read from the row (col. 13 line 56-57).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 2, 15, 19, and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sedlar as applied to claims 1, 3-12, 14, 16-17, 20-22, 24, and 26-35 above in view of Reed et al ('Reed' hereafter) (U.S. Patent 7,035,874 B1).

With respect to claim 2 and similar claims 15, 19, and 24, Sedlar fails to teach wherein the data table row that includes a user defined type corresponding to the item.

Reed, however, teaches a data table row that includes a user defined type corresponding to the item (col. 4 line 19, i.e. a MediaUDT) for including a user defined type field (Reed at col. 1 line 14-16).

In the same field of endeavor, (i.e. data control), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because Reed would have given Sedlar a user defined type field for a more robust system of including user defined data types.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert M. Timblin whose telephone number is 571-272-5627. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/ROBERT TIMBLIN/

Examiner, Art Unit 2167

/John R. Cottingham/

Supervisory Patent Examiner, Art Unit 2167